ABSTRACT

Khan, Md Ariful Ph.D., Purdue University, June 2017. Parallel Graph Algorithms through Approximation. Major Professor: Alex Pothen.

We consider the problem of computing a b-Matching and a b-Edge Cover, which are subgraphs of a graph with specific properties. Although there are polynomial-time algorithms for these problems, exact algorithms are impractical for massive graphs with billions or more of edges. Hence we design approximation algorithms that have fast run time complexity on serial computers. Since computations on massive graphs are compute-intensive, We also design approximation algorithms that possess a high degree of concurrency, so that they can be implemented efficiently on shared- and distributed-memory multiprocessors.

The b-Matching problem is a generalization of the well-known Matching problem in graphs, where the objective is to choose a subset of M edges in the graph such that *at most* a specified number $b(v)$ of edges in M are incident on each vertex v. Subject to this restriction we maximize the sum of the weights of the edges in M. We describe a half-approximation algorithm, b-Suitor, for computing a b-Matching of maximum weight in a graph with weights on the edges. Our results show that the b-Suitor algorithm outperforms previously known algorithms, the Greedy and LD (locally dominant edge) algorithms, by one to two orders of magnitude on a serial processor. The b-Suitor algorithm has a high degree of concurrency, and it scales well up to 240 threads on a shared memory multiprocessor. We also implement the algorithm in distributed memory settings using a hybrid strategy where inter-node communication uses MPI and intra-node computation is done with OpenMP threads. We demonstrate strong and weak scaling of b-Suitor up to $16K$

processors on two supercomputers at NERSC. We compute a b-MATCHING in a graph with 2 billion edges in under 4 seconds using $16K$ processors.

The b-EDGE COVER problem is a generalization of the better-known *Edge Cover* problem in graphs, where the objective is to choose a subset C of edges in the graph such that *at least* a specified number $b(v)$ of edges in C are incident on each vertex v. In the weighted b-EDGE COVER problem, we minimize the sum of the weights of the edges in C. We design three new approximation algorithms for the b-EDGE COVER problem, and compare them with the previously known GREEDY algorithm. At each step, the GREEDY algorithm updates the effective weights of the edges, and adds an edge of minimum effective weight to the current edge cover. The updates of the effective weights makes the GREEDY algorithm sequential and impractical for massive graphs. A second algorithm, the LSE (locally sub-dominant edge) algorithm, adds edges with minimum effective weight in its neighborhood to the current cover, and it is amenable for parallelization. The LSE algorithm computes the same edge cover as the GREEDY algorithm, and both are 3/2-approximation algorithms. We design a third algorithm, S-LSE, an extension of the LSE algorithm, which uses static edge weights instead of dynamic effective weights used by the latter. This relaxation causes S-LSE to have a worse approximation ratio of 2, but makes it more amenable for efficient implementation and parallelization. A fourth algorithm, the MCE (matching complement edge cover) algorithm, is obtained from a relationship between approximation algorithms for the b-EDGE COVER and the b-MATCHING problems. We prove that both S-LSE and MCE algorithms compute the same b-EDGE COVER, and hence both have approximation ratios of 2. In practice, all these algorithms compute edge covers with weights that are close to the optimal b-EDGE COVER, and have weights within 10% of each other. We parallelize and report results from the three new approximation algorithms, LSE, S-LSE and MCE in the context of shared memory multi-core machine. Our results show that the MCE algorithm is faster than the other algorithms by at least an order of magnitude, both on serial and shared memory multiprocessors.

As an application for b-Matching and b-Edge Cover, We explore the problem of sharing data with anonymity guarantees where each user defines a desired level of privacy. We provide a new algorithm for sharing data for learning purposes with all required privacy guarantees. We propose the first shared-memory parallel algorithm for the adaptive anonymity problem that achieves this goal and produces high quality anonymized datasets. We show two formulations of the adaptive anonymity problem using b-Matching and b-Edge Cover respectively. These formulations are space efficient, since it can compute a b-Matching by working with a subgraph at a time, with the size of the subgraph bounded by the number of vertices. We are able to solve anonymity problems with $500,000$ instances and a hundred features on an Intel® Xeon® multi-core processor in two hours, which are infeasible for earlier algorithms. On smaller problems, our algorithm is two orders of magnitude faster than an earlier algorithm based on Belief Propagation. Other applications for b-Matching and b-Edge Cover include graph construction from noisy data, graph sparsification, clustering, semi-supervised learning, etc.